

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INTELLIGENT SYSTEMS

## INTERAKTIVNÍ VÝUKA PRO UNIVERZITU TŘETÍHO VĚKU

BAKALÁŘSKÁ PRÁCE

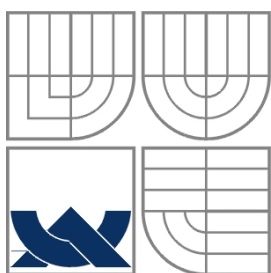
BACHELOR'S THESIS

AUTOR PRÁCE

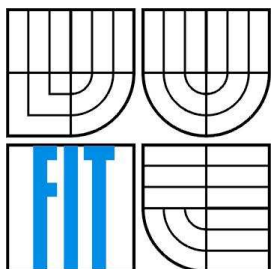
AUTHOR

ALEŠ VAVŘÍNEK

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INTELLIGENT SYSTEMS

## INETRAKTIVNÍ VÝUKA PRO U3V

U3V INTERACTIVE EDUCATION

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

ALEŠ VAVŘÍNEK

VEDOUCÍ PRÁCE  
SUPERVISOR

Ing. PAVLA SEHNALOVÁ

BRNO 2009

**Vysoké učení technické v Brně - Fakulta informačních technologií**

Ústav inteligentních systémů

Akademický rok 2008/2009

**Zadání bakalářské práce**

Řešitel: **Vavřínek Aleš**

Obor: Informační technologie

Téma: **Interaktivní výuka pro U3V**

Kategorie: Web

**Pokyny:**

1. Seznamte se s technologiemi a dostupnými nástroji pro tvorbu a vývoj aplikací, zejména PHP, MySQL, Java aj.
2. Seznamte se se současnou výukou U3V na FIT. Konzultujte požadavky na aplikaci s vyučujícím na U3V.
3. Navrhněte aplikaci na základě zjištěných požadavků a vytvořte vhodné modely interaktivního programu.
4. Zvolte vhodné implementační prostředí a koncepci realizujte pomocí prostředků dle 1. Systém naplňte učebními materiály a sadou testů.
5. Zhodnoťte výhody a nevýhody navrhovaného řešení a diskutujte možné pokračování projektu.

**Literatura:**

- Gilmore, W. J.: Velká kniha PHP5 & MySQL
- Ullman, L.: PHP a MySQL - Názorný průvodce tvorbou dynamických WWW stránek
- dle pokynů vedoucího

Při obhajobě semestrální části projektu je požadováno:

- Bez požadavků.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Sehnalová Pavla, Ing., UITS FIT VUT**

Datum zadání: 1. listopadu 2008

Datum odevzdání: 27. ledna 2009

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
Fakulta informačních technologií  
Ústav inteligentních systémů  
612 66 Brno, Božetěchova 2

---

doc. Dr. Ing. Petr Hanáček  
vedoucí ústavu

## **Abstrakt**

Tato bakalářská práce je určena pro Univerzitu třetího věku při VUT v Brně (U3V). Zejména pro její studenty, kterými jsou osoby staršího věku, kteří se chtějí i po překročení určité věkové hranice nadále vzdělávat a jít s dobou. Tato práce byla vytvořena pomocí jazyků HTML, CSS, PHP, JavaScript s použitím databáze MySQL.

## **Klíčová slova**

Informační systém, Univerzita třetího věku, U3V, studijní materiály, testy, HTML, CSS, PHP, JavaScript, MySQL.

## **Abstract**

This bachelor's thesis is designed for the University of the Third Age by VUT in Brno (U3V). Especially for their students, who are the elderly people, who like to study and to go with times over crossing the age limit. This project has been created via programming languages HTML, CSS, PHP, JavaScript using database MySQL.

## **Keywords**

Information system, University of the Third Age, U3V, study materials, tests, HTML, CSS, PHP, JavaScript, MySQL.

## **Citace**

Aleš Vavřínek: Interaktivní výuka pro U3V. Brno, 2009, bakalářská práce, FIT VUT v Brně.

# Interaktivní výuka pro U3V

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Pavly Sehnalové. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Aleš Vavřínek  
14. ledna 2009

## Poděkování

Za odbornou pomoc a vedení děkuji své vedoucí Ing. Pavle Sehnalové. A v neposlední řadě bych chtěl poděkovat všem blízkým, kteří mi pomáhali a podporovali mě.

© Aleš Vavřínek, 2009.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

Obsah .....	1
1 Úvod .....	3
2 Použité programovací jazyky .....	4
2.1 Jazyk HTML .....	4
2.1.1 Historie HTML .....	4
2.1.2 Některé použité prvky .....	5
2.2 CSS - Kaskádové styly .....	5
2.2.1 Některé použité prvky .....	5
2.3 Jazyk JavaScript .....	6
2.3.1 Historie Javascriptu .....	6
2.3.2 Některé použité prvky .....	7
2.4 Jazyk PHP .....	7
2.4.1 Historie PHP .....	8
2.4.2 Některé použité prvky .....	8
2.5 Databáze MySQL .....	9
2.5.1 Některé použité prvky .....	9
3 Specifikace a analýza požadavků .....	11
3.1 Přihlášení do systému .....	11
3.2 Registrace nových studentů .....	11
3.3 Popis jednotlivých případů užití .....	12
3.3.1 Funkce přístupné studentům .....	12
3.3.2 Funkce přístupné správcům předmětu .....	13
3.3.3 Funkce přístupné správcům systému .....	14
4 Návrh systému .....	16
4.1 Normalizace .....	16
4.1.1 1. Normální forma (1.NF) .....	16
4.1.2 2. Normální forma (2.NF) .....	16
4.1.3 3. Normální forma (3.NF) .....	17
4.2 ER – diagram .....	17
4.3 Uživatelské rozhraní .....	19
4.3.1 Vzhled .....	19
4.3.2 Kontrola chyb uživatelů .....	19
5 Implementace systému .....	21
5.1 Uživatelé .....	21

5.1.1	Student .....	21
5.1.2	Správce předmětu.....	22
5.1.3	Správce systému.....	22
5.2	Některé zajímavé funkce programu .....	23
5.2.1	Ostré testy .....	23
5.2.2	Cvičné testy.....	24
5.2.3	Přihlašování uživatelů.....	25
5.2.4	Studijní materiály.....	26
6	Testování systému.....	27
6.1	Testování použitelnosti.....	27
6.2	Testování přihlašování.....	29
6.3	Testování správné funkčnosti .....	29
7	Závěr .....	30
	Použité zkratky .....	31
	Literatura .....	32
	Seznam příloh.....	33

# 1 Úvod

Tato bakalářská práce je určena pro Univerzitu třetího věku při VUT v Brně (U3V). Zejména pro její studenty, kterými jsou osoby staršího věku, kteří se chtějí i po překročení určité věkové hranice nadále vzdělávat a jít s dobou.

V programu jsou k dispozici tři názorně vytvořené předměty s výukovými materiály a včetně sady testových otázek. Předměty jsem nezvolil zcela náhodně, ale vybral jsem takové, které umožní uživatelům systému rozhled v celé počítačové oblasti.

První předmět jsem nazval Elektronická komunikace (ECOM), kde se studenti dozvědí, co je to internet, popis prohlížeče, nebo jak napsat a číst e-mail. Druhým předmětem je Hardware (H), který poučí studenty v oblasti příslušenství počítače a stručně shrne vnitřní i vnější části computeru. Třetím předmětem je Microsoft Windows XP (WXP). Zde se nachází stručně popsane některé funkce tohoto operačního systému, které by mohli uživatelé běžně využívat.

Doufám, že tato aplikace přispěje svým dílem k lepší orientaci v počítačovém světě studentům U3V.

V druhé kapitole se zabývám použitými programovacími jazyky, které jsem pro tvorbu této aplikace využil. Zde se nachází i stručná historie těchto jazyků a ukázky použitých zdrojových kódů. Kromě programovacích jazyků je tu i stručný popis použité databáze MySQL.

Třetí kapitola pojednává o specifikaci a analýze zadaných požadavků. Řeším zde, jak se budou registrovat noví uživatelé, jak se zaregistrovaní uživatelé mohou přihlašovat do systému a také jaké funkce jsou přístupné uživatelům s různým oprávněním. K znázornění těchto funkcí jsem zde uvedl i digram případů užití.

Čtvrtá kapitola se zabývá návrhem systému. Popisuji zde, jak se postupně normalizuje navrhnutý systém a s příklady na vývoji této aplikace jednotlivé normální formy. Jako výsledek normalizace zde uvádím i vytvořený E-R diagram. Zbytek kapitoly je věnován uživatelskému rozhraní systému včetně popisu vzhledu a popisu kontrolování chyb uživatelů.

Pátá kapitola je věnována implementaci systému. Jsou tu detailně popsány všechny typy uživatelů systému. U každého typu je popsáno jeho oprávnění a funkce, které může využívat. Následující část je věnována některým zajímavým naprogramovaným funkcím. Popisuji zde princip funkce ostrých i cvičných testů, přihlašování uživatelů a studijní materiály.

Předposlední kapitola je věnována testování systému. Po ukončení programování, vložení studijních materiálů a vytvoření sady testů pro každý předmět jsem mohl začít testovat systém na lidech. Jsou zde vyobrazeny výsledky, jak se jim podařilo systém ovládat a jak se jim dařilo při ostrém testu. Krom těchto informací jsem zde shrnul zabezpečení přihlašování a funkčnost celé aplikace.

V poslední kapitole jsem shrnul dosažené výsledky a naznačil možná rozšíření.



## 2 Použité programovací jazyky

Tato kapitola se věnuje použitým programovacím jazykům, které se v mojí bakalářské práci vyskytly. Jedná se o jazyky používané pro implementaci dynamických webových aplikací s využitím databázového přístupu.

### 2.1 Jazyk HTML

Hypertext Markup Language (dále jen HTML) je značkovací jazyk pro vytváření statických WWW stránek. Aktuálně používaná verze, schválená konsorciem W3C (The World Wide Web Consortium) je 4.01.

#### 2.1.1 Historie HTML

Pokud chceme vědět něco o historii HTML, musíme se nejprve zaměřit na jeho předka SGML (Standard Generalized Markup Language). V roce 1986 byl schválen organizací ISO (International Organization for Standardization). Dokumenty SGML obsahují text, multimediální prvky a mohou obsahovat nadpisy všech úrovní, odstavce a formátovací prvky. Protože je velice složitý, moc se v internetu nerozšířil. Jeho velkou předností je možnost definovat nové značky (tagy). Proto se používá v současné době pro vývoj nových značkovacích jazyků. Ty jsou vždy vytvářeny pomocí DTD (Definice Typu Dokumentu). DTD definuje formátovací značky, jejich vlastnosti a hodnoty, kterých mohou nabývat.

Nyní se již dostáváme k samotnému HTML, který slouží k vytváření dokumentů. Tyto dokumenty mohou obsahovat hypertextové odkazy a pokročilejší formátování. HTML vznikl v roce 1990 jakožto verze HTML 1.0, ale v roce 1993 byl rozšířen verzí HTML + (obsahoval 78 elementů). Nová verze HTML 2.0, která byla podstatně zjednodušena (obsahovala 49 elementů) vznikla v roce 1994. Ta se jako první stala i uznávaným standardem. V roce 1995 se objevila nová verze HTML 3.0, což byl jakýsi kompromis mezi HTML + a HTML 2.0. Objevily se tu tabulky, matematické elementy, možnost formátování, formuláře. Přes verzi 3.2 v roce 1996 se plynule přešlo k nové verzi HTML 4.0, která se používá dnes. Tato verze umožňuje internacionalizace (používání zvláštních znaků různých abeced), úpravy tabulek a formulářů. Velký přínos byla podpora kaskádových stylů, na které se zaměříme níže. [3]

## 2.1.2 Některé použité prvky

Důležitou součástí práce jsou formuláře:

Např.:

```
<form id="menuForm" name="menuForm" method="post" action="login.php">
<input type="hidden" name="login" value="".$login.">
<input type="hidden" name="heslo" value="".$heslo.">
<input type="hidden" name="predmet" value="".$predmet.">
<input type="hidden" NAME="str" value="0">
<button type="submit" id="menu-b1" value="1" onclick="menuForm.str.value=1;"><label>
STUDENTI </label></button><br>
...
</form>
```

Popis: Formulář *menuForm* zobrazuje rozcestník s odkazy na jiné stránky přístupné s daným oprávněním. Využívá skrytých (*hidden*) proměnných s uloženou hodnotou, předávanou pomocí jazyka PHP, pro odesílání dříve zvolených informací. Odesílací tlačítko využívá JavaScriptové funkce pro odeslání hodnoty při kliknutí (na pohled složitější implementace, ale funguje jak v prohlížeči Firefox tak i Explorer). Metoda odesílání dat je *post* – odesílaná data nejsou zobrazena v prohlížeči v poli adresa.

## 2.2 CSS - Kaskádové styly

Cascading Style Sheets (CSS) slouží k popisu prezentace dokumentů, aniž by jakkoli ovlivňovali jejich obsah a strukturu. Jedná se vlastně o externí soubor, který určuje vzhled elementů v HTML souboru.

Před nástupem kaskádových stylů bylo ve značkovacích jazycích (např. HTML) pro potřeby formátování a pozicování prvků na stránce používáno tabulek a několikanásobně vnořených tabulek, což vedlo k tomu, že většina obsahu zdrojového kódu se zabývala především vzhledem nikoli obsahem.

CSS byl navržen organizací W3C. Autor prvotního návrhu byl Håkon Wium Lie. Publikoval jejich koncept v roce 1994. Prozatím byly vydány dvě verze – CSS1 a CSS2. Nyní se dokončuje verze CSS 2.1 a pracuje se na nové verzi CSS3.

Nevýhoda kaskádových stylů je různá reprezentace stejného CSS kódu v různých webových prohlížečích. Nástupem nových verzí prohlížečů se ale situace výrazně zlepšuje. [4], [13]

### 2.2.1 Některé použité prvky

Většina formátovacích značek se nachází v externím dokumentu *styl.css* pro zpřehlednění vlastního HTML a PHP kódu.

Např.:

```
body {background: lightyellow;}
body {background-image: url("pozadi.jpg"); background-repeat: repeat-x; background-attachment: fixed;}
```

```
/*nadpis a horni logo*/
div.nadpis-u3v {font-family: Algerian; font-size: 40pt; font-style: italic; text-align: center;}
div.nadpis-u3v img {float:left; margin-top: -0.22em;}
```

```
/*rozcestnik*/
div.rozcestnik {margin-top: 5em; border-width: 5px; border-style: solid; border-color:
rgb(1,113,187); float: left; margin-left: -10em;}
#b1 {text-align: center; background: lightblue; height: 40px; width: 230px;}
#b2 {text-align: center; background: white; height: 40px; width: 230px;}
```

...

Popis: Část souboru styl.css. Popisuje zobrazení pozadí všech stránek, záhlaví a rozcestník na úvodní stránce.

## 2.3 Jazyk JavaScript

JavaScript je multiplatformní objektově orientovaný skriptovací jazyk požívaný v prostředí WWW (World Wide Web) pro tvorbu aplikací. Obvykle se skript v tomto jazyce vkládá přímo do kódu HTML stránky. Obvykle jsou jím ovládány interaktivní prvky jako např. tlačítka nebo textová pole.

Syntaxe jazyka patří do rodiny C/C++/Java. I když název napovídá, že jde o verzi Javy, není tomu tak a společně mají vedle podobného názvu jen podobnou syntaxi.

Program v JavaScriptu se obvykle spouští po stažení WWW stránky z internetu = na straně klienta na rozdíl od jiných programovacích jazyků (např. PHP), které se spouštějí na serveru. Z toho vyplývá, že JavaScript nemůže například pracovat se soubory, aby tím neohrozil bezpečnost klienta.

[5], [9]

### 2.3.1 Historie Javascriptu

Byl vyvinut v roce 1995 firmou Netscape nejdříve pod názvem Mocha, potom LiveScript a nakonec JavaScript. Firma Microsoft používá název JScript.

V roce 1997 byl standardizován asociací ECMA (European Computer Manufacturers Association) a v srpnu 1998 i organizací ISO. [9]

### 2.3.2 Některé použité prvky

Tento jazyk jsem hojně využil pro odesílání dat z HTML formulářů (viz výše) anebo například v ostrých testech na odpočítávání času do konce testu.

Např.:

```
if($timeout == NULL){$timeout = $TIMEOUT;}
echo '
<script type="text/javascript">
    odpocitavani= "";
    timeout='.$timeout.';

    function odpocet(){
        if(timeout>0){
            timeout -= 1;
            min=Math.floor(timeout/60);
            sec=timeout % 60;
        }
        odpocitavani = min + " minut " + sec + " sekund";
        document.otazkyForm.timeout.value=timeout;
        document.odpocet.cas.value = odpocitavani;
        if(timeout == 0){
            alert("konec testu!");
            timeout=-1;
            document.otazkyForm.timeout.value=timeout;
            self.document.forms.otazkyForm.submit();
            self.document.forms.odpovediForm.submit();
        }
    }
</script>

<form name="odpocet">
    Čas do konce testu:
    <input type="text" name="cas" id="cas" disabled size="15">
    <script language="JavaScript">
        setInterval("odpocet()",1000);
    </script>
</form>';
```

Popis: funkce pro odpočítávání času v ostrém testu. Pokud je stránka aktualizována, použije se již uplynulý čas. Aktuální čas do konce se zobrazuje do textového formulářového políčka. Funkci voláme každou vteřinu (1000 milisekund). Po uplynutí času se odešlou vyplněné odpovědi do souboru s vyhodnocením testu.

## 2.4 Jazyk PHP

PHP je skriptovací jazyk určený především pro programování dynamických internetových stránek. Používá se jako proveditelný kód, vložený do HTML stránek. Skripty jsou většinou prováděny na

straně serveru. Programátor má možnost s pomocí PHP se připojit k databázi (podpora mnoha typů databází včetně MySQL, kterou ve své práci používám).

PHP se stalo velmi oblíbeným především díky jednoduchosti použití a tomu, že kombinuje vlastnosti více programovacích jazyků a nechává tak vývojáři částečnou svobodu v syntaxi. V kombinaci s operačním systémem Linux, databázovým systémem (obvykle MySQL nebo PostgreSQL) a webovým serverem Apache je často využíván k tvorbě webových aplikací. Pro tuto kombinaci se vžila zkratka LAMP – tedy spojení Linux, Apache, MySQL a PHP nebo Perl. [1], [2]

## 2.4.1 Historie PHP

PHP vyvinuli Rasmus Lerdorf (tvůrce), Zeev Suraski a Andi Gutmas v roce 1994 jako Personal Home Page. V polovině téhož roku se systém PHP spojil s programem Form Interpreter a tak vzniklo PHP/FI 2.0. V roce 1997 přepsali parser a zformovali tak základ PHP3. Tehdy byl také změněn název na Hypertext Preprocessor, který je platný i dnes. Verze 3 vyšla v roce 1998. Byla rychlejší a obsahovala více funkcí. Navíc bylo možné tuto verzi spustit i na operačním systému Windows (především verze fungovali pouze na OS Linux). Verze 4 vyšla v roce 2000.

Od roku 2004 se používá verze 5 s vylepšenou podporou objektově orientovaného programování, které se inspirovalo jazykem Java. Aktuálně nejnovější verze je PHP 5.2.6 z 1. května 2008. [10], [11]

## 2.4.2 Některé použité prvky

Jazyk PHP je využit na operace spojené s přístupem do databáze, proměnné ve formulářích, výpisy tabulek z databáze.

Např.:

```
1.
//kontrola zda je login v databázi
mysql_select_db('czprojekt-u3v');
$l=mysql_query("select * from osoba where login = '$login'");
if (!mysql_num_rows($l)){ //není, uživatel nemůže být vpuštěn do systému
    include("index.php");
    echo "<font size='5' color='red'>";
    echo "Neznámý login!!!<BR>";
    echo "</font>";
    echo "<font size='5' color='green'>";
    echo "Možné příčiny:<BR>";
    echo "1) nejste v databázi doposud zaregistrováni<BR>";
    echo "2) zadali jste svůj login špatně<BR>";
    echo "</font>";
}
```

2.

```
$j=mysql_query("select jmeno from osoba where login = '$login'");
$p=mysql_query("select prijmeni from osoba where login = '$login'");
$jmeno = mysql_fetch_object($j);
$prijmeni = mysql_fetch_object($p);
echo '<div class="logged">';
echo "<u>přihlášen</u>: <b>student</b> <br>";
echo "<b>$jmeno->jmeno </b>";
echo "<b>$prijmeni->prijmeni</b><br>";
echo "</div>";
```

Popis:

1. Kontrola, zda je zadané přihlašovací jméno v databázi. Pokud není, nemůže být uživatel vpuštěn do systému a bude mu tato informace vypsána na hlavní stránku.
2. Pokud je uživatel přihlášen, zobrazují se v záhlaví stručně jeho údaje a možnost odhlášení ze systému.

## 2.5 Databáze MySQL

MySQL je databázový systém, vytvořený švédskou firmou MySQL AB. Hlavní autoři jsou Michael Widenius a David Axmark. MySQL je multiplatformní databáze (může být implementována na různých operačních systémech (Linux, Windows,...)). Spojení klient-server je optimalizováno především na rychlost i přes některá zjednodušení. Dalším kladem je možnost volného šíření databáze. Díky těmto vlastnostem je MySQL jedna z nejvíce používaných databází.

Komunikace s databází probíhá pomocí jazyka SQL (Standard Query Language). [\[1\]](#), [\[12\]](#)

### 2.5.1 Některé použité prvky

**Spouštění ostrého testu** – ukázka kódů obsahuje i prvky jazyka PHP. Jak jsem již psal výše v kapitole o tomto jazyce, tomuto spojení se vžila zkratka LAMP – tedy spojení Linux, Apache, MySQL a PHP nebo Perl. [\[2\]](#)

1. Zjistíme čísla, názvy a id testů, přístupných studentům pro vybraný předmět:
  - *select cislo\_testu, nazev\_testu, test\_id from testy where predmet\_id="\$.predmet." and pristupny\_studentum="1" order by cislo\_testu*

2. Student vybral test číslo 1. Zjistíme, zda ho student už nepsal:

- *select test1 from studuje where student="\$.\$rodneCislo." and predmet="\$. \$predmet."*

3. Pokud je test1 0, přepíšeme ji na -1:

- *update studuje set test1="-1" where student="\$. \$rodneCislo." and predmet="\$. \$predmet."*

4. Zjistíme výsledky všech testů a sečteme body a zjistíme dnešní datum:

- *select test1, test2, test3, test4, test5 from studuje where student="\$. \$rodneCislo." and predmet="\$. \$predmet."*
- *\$poc=\$pocet\_bodu["test1"] + \$pocet\_bodu["test2"] + \$pocet\_bodu["test3"] + \$pocet\_bodu["test4"] + \$pocet\_bodu["test5"];*
- *\$DnesniDatum=date("Y-m-d");*

5. Zjistíme body potřebné na zápočet a vyhodnotíme udělení zápočtu:

- *select zapocet from predmety where id="\$. \$predmet."*
- *\$zap=mysql\_fetch\_object(\$z);*  
*if(\$poc>=\$zap->zapocet){\$zapocet="ano";}*  
*else if(\$poc<\$zap->zapocet){\$zapocet="ne";}*  
*else if(\$zap->zapocet==0){\$zapocet="";}*  
*else \$zapocet="??";*

6. Nakonec vypočítáme výslednou známku a upravíme údaje v tabulce:

- *if(\$poc<50){\$znamka="4F";}*  
*else if(\$poc>=50 && \$poc<60){\$znamka="3E";}*  
*else if(\$poc>=60 && \$poc<70){\$znamka="3D";}*  
*else if(\$poc>=70 && \$poc<80){\$znamka="2C";}*  
*else if(\$poc>=80 && \$poc<90){\$znamka="2B";}*  
*else if(\$poc>=90 && \$poc<=100){\$znamka="1A";}*  
*else \$znamka="??";*
- *update studuje set body="\$. \$poc.", datum="\$. \$DnesniDatum.", znamka="\$. \$znamka.", zapocet="\$. \$zapocet." where student="\$. \$rodneCislo." and predmet="\$. \$predmet."');*

## **3 Specifikace a analýza požadavků**

Univerzita třetího věku (U3V) při VUT v Brně má požadavek vytvořit informační systém, který bude sloužit pro výuku. Systém bude umožňovat jednoduchou správu uživatelů, vkládání výukových materiálů a testů.

V tomto systému se budou nacházet tři typy uživatelů – hlavní správce systému, správce předmětu a student.

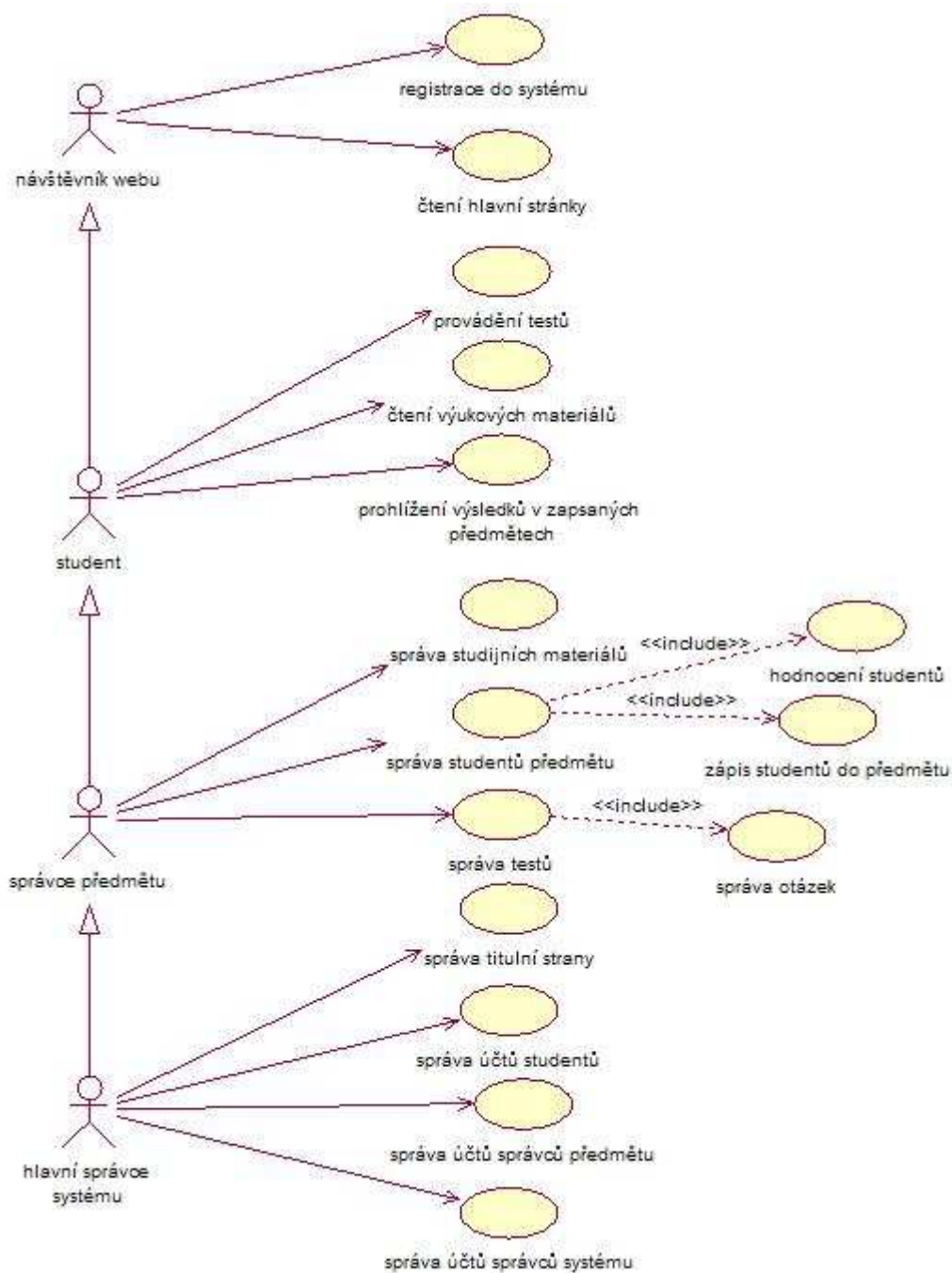
### **3.1 Přihlášení do systému**

Pro přihlášení se bude používat základní obrazovka a bude stejná pro všechny typy uživatelů. Pro přihlášení do systému bude nutné zadat přihlašovací jméno a heslo. Na základě vyplněných údajů systém rozpozná oprávnění uživatele a načte příslušnou úvodní stránku. Při zadání špatných údajů nebude uživateli umožněn vstup do systému a bude tato skutečnost vypsána na obrazovku.

### **3.2 Registrace nových studentů**

Noví studenti se budou moci registrovat buď na hlavní stránce přes sekci E-přihláška ke studiu, nebo přímo kontaktovat správce systému, který je zaregistruje. V prvním případě se nový student registruje přímo do systému, ale jeho účet musí před možností přihlášení vždy nejprve aktivovat správce systému.





Obrázek 1: Diagram případů užití

### 3.3 Popis jednotlivých případů užití

(viz [Obrázek 1](#))

#### 3.3.1 Funkce přístupné studentům

Po přihlášení studenta se načte jeho osobní stránka s výpisem předmětů, které bude mít zapsané. Pro zjednodušení práce se systémem budou mít studenti všechny potřebné informace na jedné stránce, aby je nemuseli složitě hledat. Po zvolení konkrétního předmětu bude mít student možnost:

### **1. Prohlížení výsledků předmětů**

Zobrazí se podrobnější hodnocení předmětu – výsledky testů, popř. zkoušky. Na úvodní stránce se zobrazují všechny předměty, které bude mít student zapsané.

### **2. Čtení studijních materiálů**

Výukové materiály budou vkládány a editovány správcem předmětu. Student bude mít možnost pouze jejich čtení. Studijní materiály se budou zobrazovat vždy pro zvolený předmět. Do studijních materiálů bude moci správce předmětu také vkládat např. termíny zkoušek, doporučenou literaturu, nebo jiné aktuality k předmětu.

### **3. Provádění cvičných testů**

Cvičné testy budou sloužit k procvičení probírané látky. Po vybrání konkrétního testu si student bude moci projít několik otázek cvičně.

### **4. Provádění ostrých testů**

Ostré testy budou sloužit k otestování znalostí studentů a na jejich základě potom vyučující bude přidělovat studentům body do předmětu. Budou se psát pouze v učebně za kontroly vyučujícího. Po skončení testu se studentům automaticky zapíše body zaokrouhlené nahoru (1,01 se zaokrouhlí na 2) do informačního systému. Vyučující obejde studenty a případně jim vysvětlí chyby. Ostré testy bude nastavovat správce předmětu (délku testu, počet otázek, počet bodů za test). Aby si studenti nemohli zkoušet ostré testy doma, zpřístupní je správce předmětu až před začátkem testu a potom, co si všichni studenti test zapnou, ho může opět znepřístupnit.

## **3.3.2 Funkce přístupné správcům předmětu**

Po přihlášení si správce předmětu vybere předmět, který bude právě spravovat. Po zvolení předmětu bude mít následující možnosti:

### **1. Správa studentů předmětu**

Správce bude mít možnost zapisovat a odepisovat studenty a dále měnit hodnocení zapsaných studentů. Zapisovat může pouze ty studenty, kteří mají již aktivní účet (aktivovaný správcem systému).

### **2. Správa testů**

Bude moci vkládat nové testy a upravovat již vytvořené. Pokud nebude chtít přidat nový test, vybere z nabídky již vložených testů pro správu konkrétního testu. Nyní bude mít možnost zkontrolovat otázky, které test obsahuje; vkládat nové otázky; upravovat otázky; mazat otázky; měnit nastavení testu. Nastavení bude obsahovat: název testu, číslo testu, počet

otázek, dobu trvání testu, počet bodů za test a zpřístupnění studentům. Při vytváření otázek nebude zadávat správce přímo body, ale zadá obtížnost otázky. Podle počtu bodů za test se potom body rozdělí podle obtížnosti k jednotlivým otázkám.

### **3. Správa studijních materiálů**

Správce bude moci přidávat a editovat studijní materiály pro předmět, který spravuje. Správa bude fungovat editací HTML kódu přímo přes webové rozhraní. Každý předmět bude mít vlastní HTML stránku uloženou na serveru. Pro nové předměty se bude soubor automaticky vytvářet při prvním vkládání informací k předmětu. Tady předpokládám, že správcové předmětu zvládají psaní jednoduchých webových dokumentů.

## **3.3.3 Funkce přístupné správcům systému**

Po přihlášení správce systému se načte stránka s výběrem oprávnění. Bude mít totiž možnost se přihlásit buď jako správce libovolného předmětu (pro možnost kontroly předmětů), nebo jako správce systému. Pokud si zvolí správu předmětu, bude moci k němu přistupovat úplně stejně, jako by byl správce daného předmětu. Pokud si zvolí přihlášení jako správce systému, tak bude mít následující možnosti:

### **1. Správa studentů**

Správce systému na rozdíl od správce předmětu bude mít oprávnění k výpisu všech studentů; zápisu nového studenta; editaci údajů o studentech; mazání studentů; výpisu neaktivních studentů (studenti, kteří nemají zapsaný žádný předmět) a bude mít možnost aktivovat účet studentům, kteří se registrovali přes hlavní stránku systému (odkaz E-přihláška ke studiu). Pokud správce systému nebude chtít daný účet aktivovat, bude jej moci rovnou jednoduše odstranit.

### **2. Správa zaměstnanců**

Správa zaměstnanců bude mít stejné možnosti jako správa studentů, ale navíc se zde budou moci měnit oprávnění jednotlivých zaměstnanců. Správce bude moci měnit údaje všem zaměstnancům včetně sebe. Nebude se ale moci smazat. Zaměstnanci se nebudou moci registrovat přes přihlášku ke studiu. Oprávnění k registraci zaměstnanců bude mít pouze správce systému.

### **3. Správa předmětů**

Zde bude moci správce přidávat nové předměty a vybírat pro ně správce; měnit údaje o předmětu (předmět bude mít zkratku, název, typ, ukončení, počet kreditů, body potřebné na zápočet, počet testů a 1 nebo 2 správce předmětu); mazání předmětu.

#### 4. **Správa titulní strany**

V této sekci bude upravovat správce systému titulní stranu systému (úvodní obrazovku). Na této stránce bude elektronická přihláška ke studiu, dny otevřených dveří, přijímací řízení, studijní programy, kontakty, informace o U3V, poplatky, umístění U3V na mapě, ukázka testu a aktuality. Každý z těchto odkazů bude HTML stránka, kterou bude moci správce přímo přes webové rozhraní editovat. Tady opět předpokládám, že i správcové systému zvládají psaní jednoduchých webových dokumentů.

## 4 Návrh systému

Po zpracování a analýze požadavků se nyní zaměřím na návrh systému. Z výše zmíněných požadavků vytvořím E-R diagram a návrh uživatelského rozhraní.

### 4.1 Normalizace

Normalizací se rozumí odstranění redundantních dat (data, které se opakují), omezení složitosti (rozdělení příliš velké tabulky na více celků) a zabránění tzv. aktualizacím anomáliím (pokud smažeme všechny předměty, které učí daný správce předmětu, nechceme, aby se smazala z databáze data o této osobě). Normalizace by měla vést k přehlednější a výkonnější databázi. [\[14\]](#)

#### 4.1.1 1. Normální forma (1.NF)

Relace je v první normální formě, pokud každý její atribut obsahuje jen atomické hodnoty. Tedy hodnoty z pohledu databáze již dále nedělitelné. [\[14\]](#)

Například student U3V bude chtít studovat více předmětů. Mohly by se uložit do tabulky osoba do sloupce předměty a oddělovaly by se středníkem nebo čárkou. Toto řešení by bylo značně neefektivní ať už na vyhledávání, tak i na vkládání a další operace. Aby byla tabulka v 1.NF, musíme buď rozdělit předměty do více kategorií, což by nejspíš v tomto případě nebylo možné. Proto použijeme druhý způsob - rozdělit tuto tabulku na více tabulek. Já jsem pro tuto konkrétní skutečnost zvolil vytvoření tabulky studuje, kde se nachází zapsané předměty studenta.

Druhým příkladem je rozdělení telefonních čísel uživatelů do třech kategorií (domů, do práce, mobil).

#### 4.1.2 2. Normální forma (2.NF)

Relace se nachází v druhé normální formě, jestliže je v první normální formě a každý neklíčový atribut je plně závislý na primárním klíči, a to na celém klíči a nejen na nějaké jeho podmnožině. [\[14\]](#)

Z této definice nám vyplývá, že aby se tabulka nacházela v 2.NF, nesmí mít vícehodnotový primární klíč. Řešením je většinou rozdělením tabulky na více tabulek.

Například pokud bychom měli data o studentech pouze v tabulce studuje, tak pokud bychom studenta vyřadili ze studia všech předmětů (zrušili zápis ze všech předmětů), ztratili bychom údaje o této osobě.

### 4.1.3 3. Normální forma (3.NF)

V této formě se nachází tabulka, splňuje-li předcházející dvě formy a žádný z jejích atributů není tranzitivně závislý na klíči. Jiné vyjádření téhož říká, že relace je v 3.NF, pokud je ve 2.NF a všechny neklíčové atributy jsou navzájem nezávislé. [14]

Například v tabulce osoba máme údaje o studentech a zaměstnancích U3V. Součástí těchto údajů je i informace o adrese. Podle 3.NF by neměli být žádné neklíčové atributy vzájemně závislé. Tato tabulka porušuje definici tím, že se tu nachází město i PSČ, které jsou neklíčové atributy a jsou na sobě závislé. Řešením by bylo rozdělení této tabulky na tabulku osoba, kde by bylo ID města a tabulku adresa, kde by se nacházela daná města s PSČ. Toto rozdělení se mi v tomto případě zdálo nevhodné, proto jsem tuto tabulku nechal pouze v 2.NF. Ostatní tabulky se nacházejí v 3.NF.

## 4.2 ER – diagram

Model uložení dat v databázi se zobrazuje pomocí [Entity Relationship diagramu](#).

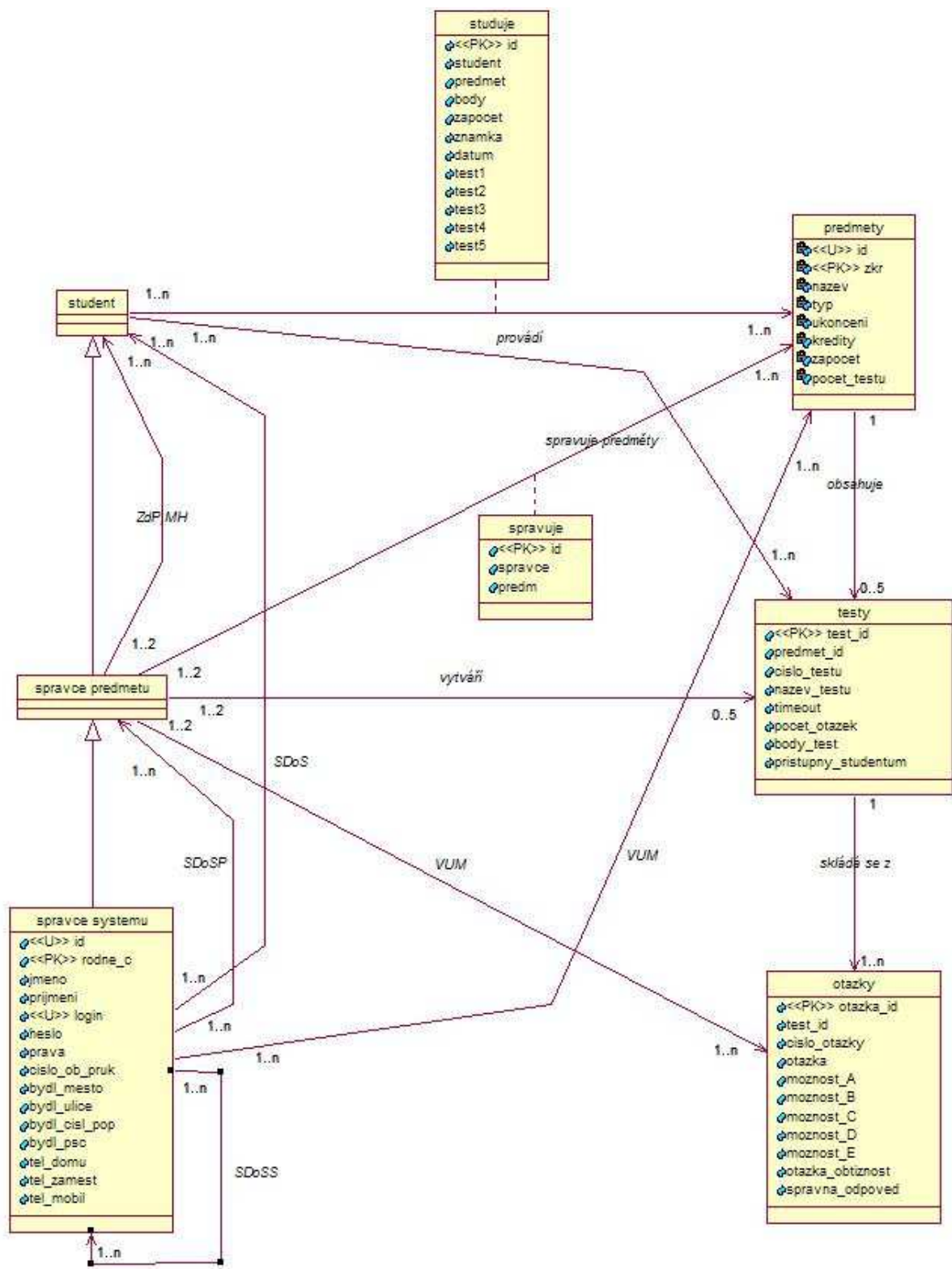
Vysvětlení k diagramu – třídy student, správce předmětu a správce systému jsou ve skutečnosti uloženy dohromady v tabulce osoba. Rozdělení jsem zde provedl pouze pro rozlišení funkcí těchto uživatelů. V tabulce osoba se uživatelé rozlišují pouze podle sloupce „prava“. 0 – student, 1 – správce předmětu, 2 – správce systému.

- **Student** – má přístup pouze do předmětů, resp. do testů předmětů, které má zapsané správcem předmětu. Zápis a výsledky testů se ukládají do tabulky „studuje“.
- **Správce předmětu** – má přístup do předmětů, které mu správce systému přiřadil. Přiřazení předmětu konkrétnímu správci se ukládá do tabulky „spravuje“. V těchto předmětech upravuje testy a otázky. Má oprávnění k vytváření, mazání a úpravě testů. Ty samé činnosti může provádět i s jednotlivými otázkami.
- **Správce systému** – má přístup do celého systému. Může měnit, přidávat a mazat data všech uživatelů systému. Stejně tak může i měnit, přidávat a mazat předměty.
- **Předměty, testy, otázky** – předměty obsahují testy a testy zase obsahují testové otázky.

#### Význam zkratk

- **ZdP, MH** – zapisuje do předmětu, mění hodnocení
- **SDoS** – spravuje data o studentech
- **SDoSP** – spravuje data o správcích předmětu
- **SDoSS** – spravuje data o správcích systému
- **VUM** – vytváří, upravuje, maže

Studijní materiály nejsou v tomto diagramu zobrazeny, protože nejsou uloženy v databázi.



Obrázek 2: E-R diagram

## 4.3 Uživatelské rozhraní

Uživatelské rozhraní slouží pro komunikaci uživatele s aplikací. Umožňuje užívat všechny naprogramované funkce, které má konkrétní uživatel (dle oprávnění) přístupné.

Tato aplikace bude sloužit pro Univerzitu třetího věku při VUT v Brně (U3V), proto je i uživatelské rozhraní maximálně zjednodušeno pro jednoduché a intuitivní ovládání.

### 4.3.1 Vzhled

Pro všechny uživatele je vytvořený obdobný vzhled všech stránek. Po přihlášení se uživateli objeví stránka s veškerými možnostmi použití systému s daným oprávněním. Podle mě je tato stránka na pohled velmi jednoduchá, proto obsluhu a využití systému zvládne každý.

Zvolil jsem barevné schéma, které zbytečně uživatele nedráždí. Světle žlutá a světle modrá barva působí na pohled velmi vyrovnaným a uklidňujícím dojmem.

#### 4.3.1.1 Výukové materiály

Podle mého názoru je vhodně graficky vytvořená část pro čtení výukových materiálů. Na této stránce se objevuje i kombinace se světle zelenou barvou, která zde příliš nevyniká, ale příjemně doplňuje vzhled stránky. Záměrně jsem použil větší velikost písma, protože předpokládám, že by uživatelé staršího věku, jímž je aplikace určena, měli se čtením malých písmen problémy. Studijní materiály jsou členěny na kapitoly, jejichž název je vždy tučně zvýrazněn. Celý text je členěn na odstavce a zarovnan do bloku. Domnívám se, že díky tomuto členění je text přehlednější a jasně z něj vyplívá, co k čemu patří. Pro lepší orientaci jsem důležité pojmy ztučnil.

### 4.3.2 Kontrola chyb uživatelů

- Nový uživatel se nemůže registrovat se stejným rodným číslem nebo přihlašovacím jménem jako už má jiný registrovaný uživatel systému.
- Při spouštění ostrého testu se kontroluje, kolik bodů za konkrétní test student obdržel. Pokud student test ještě nepsal, tak má 0 a je mu umožněno zahájení testu (při jakémkoliv jiné hodnotě je mu oznámeno, že na test nemá více pokusů). Zde mohou nastat chyby. Již při spouštění testu se studentovi jako výsledek zapíše -1. Toto je pouze proto, když by student vypnul test před vyhodnocením. Pokud test projde a jeho celkový výsledek je 0 bodů, tak se mu zapíše -1 bod. Pokud má jakýkoliv jiný výsledek, je vše v pořádku.
- Při psaní testu by nemělo dojít k vynechání otázky, protože se stránka automaticky posouvá dolů po jednotlivých otázkách.



Chyba může vzniknout pouze překlepem, proto se u všech zásadních akcí systém zeptá na potvrzení této akce. Toto se týká hlavně administrátorských funkcí.

- Odstranění uživatelů z databáze
- Odstranění předmětů
- Odstraňování jednotlivých otázek z testu.
- Po odstranění všech otázek se systém automaticky zeptá na odstranění testu.

## 5 Implementace systému

Po skončení fáze návrhu systému jsem začal vytvářet informační systém pro U3V. Tento systém je implementován jako webová aplikace s využitím databázového přístupu. Použil jsem všechny programovací jazyky, které jsem uvedl v druhé kapitole - HTML, CSS, JavaSkript a PHP ve spojení s databází MySQL.

Nastavení připojení k databázi je nastaveno v souboru `conect.php`. Přihlašování probíhá pomocí vytvořeného účtu na webu `www.php5.cz`, který jsem využil pro umístění testovací verze informačního systému. Naopak pro odhlášení ze systému se používá soubor `logout.php`.

Pro nastavení vzhledu aplikace jsem využil kaskádových stylů. Popis vzhledu jednotlivých tlačítek, formulářů a vzhledu stránek je uložen v souboru `styl.css`.

Krom těchto souborů, obsahuje systém množství dalších zdrojových souborů s funkcemi jako kontrola přihlašovaných uživatelů, hlavní menu pro všechny typy uživatelů, editace účtů studentů a zaměstnanců, editace testů a s tím související soubory pro vypsání otázek v cvičném i ostrém testu a vyhodnocení ostrého testu. Samozřejmě je nedílnou součástí i `index.php`, což je startovní stránka celého systému. Pro některé ikony v systému jsem využil externí obrázky. Pokud se načte hlavní stránka s chybnou diakritikou, je to proto, že je v kódování UTF-8. Stačí jen v prohlížeči stránku aktualizovat.

### 5.1 Uživatelé

Každý registrovaný uživatel má svůj uživatelský účet se základními informacemi o sobě. V databázi jsou uloženy údaje o bydlišti, osobní údaje, telefonní čísla a hlavně oprávnění konkrétního uživatele a přístupové přihlašovací jméno a heslo.

Všichni uživatelé se přihlašují do systému přímo z hlavní stránky systému. Neplatné přihlášení má několik typů – neznámé přihlašovací jméno, špatné heslo nebo neaktivní účet uživatele.

#### 5.1.1 Student

Budoucí studenti se mohou registrovat přes hlavní obrazovku systému v sekci E-Přihláška ke studiu. Takto registrovaní studenti nemají ovšem zatím do systému přístup a při pokusu o přihlášení je jim sdělena informace: Váš uživatelský účet nebyl zatím aktivován správcem!

Potom, co je studentovi účet aktivován, může se přihlásit a prohlížet svoje zapsané předměty, k nim související studijní materiály a cvičné testy. Ostré testy povolí správce předmětu vždy jen těsně před testem.

### 5.1.2 Správce předmětu

Správce předmětu se do systému přes hlavní stránku neregistruje, ale registruje se správcem systému přímo. Také každému správci předmětu přiděluje předměty, které bude vyučovat s tím, že jeden předmět mohou spravovat až 2 různí správci předmětu.

Tito uživatelé mají oprávnění k zápisu studentů do jejich předmětů jednoduchým kliknutím a ke změně hodnocení zapsaných studentů, které se řeší na samostatné stránce. Druhá část správy předmětu souvisí se správou testů. Zde se vytváří, mění a mažou testové otázky pro vytvořené testy. Správce může přidávat nové testy, nebo naopak při vymazání všech otázek mu systém automaticky nabídne smazání testu. Nastavení testu se týká pouze ostrých testů:

- název testu – má pouze informační charakter
- číslo testu – odpovídá i číslu testu ve studentových testech
- délka testu – počet sekund na vypracování
- počet otázek – rozmezí mezi 1 a celkovým počtem vložených otázek pro konkrétní test
- počet bodů – celkový maximální počet bodů za test
- přístupnost studentům – správce zpřístupní test studentům až před začátkem testu a potom jej opět znepřístupní, aby si jej studenti náhodně nespustili doma

### 5.1.3 Správce systému

Při přihlášení si správce vybírá oprávnění – pokud se chce podívat na některý předmět, vybere možnost „Přihlásit se jako správce předmětu“ a po té vybere konkrétní předmět. Po vybrání předmětu má naprosto stejné možnosti jako-by byl správce tohoto předmětu. Pokud zvolí možnost „Přihlásit se jako správce systému“, pak může organizovat celý informační systém U3V. Popíše zde funkce pro druhou volbu, protože správce předmětu je již popsán výše.

Základní funkce, které má tento uživatel na výběr:

- Studenti
  - výpis všech registrovaných studentů
  - zápis nového studenta – používá se, pokud se student předem neregistroval přes úvodní stránku
  - update studenta – změna jakékoliv položky ze sekce základní informace, bydliště, telefonů a přihlašovací údaje
  - odstranit studenta – odstranění studentů, kteří již na U3V nestudují
  - výpis neaktivních studentů – výpis studentů, kteří nemají zapsaný žádný předmět
  - výpis nově registrovaných studentů – studenti, kteří se registrovali přes E-Přihlášku ke studiu na hlavní stránce a čekají na aktivaci účtu, nebo odstranění ze systému

- Zaměstnanci – funkce, které jsou k dispozici, jsou stejné jako u správy studentů. Některé by však neměli smysl.
  - výpis všech zaměstnanců
  - zapsat nového zaměstnance
  - update zaměstnance
  - odstranit zaměstnance
- Předměty
  - výpis všech předmětů – zde se vypisují předměty s přiřazenými správci
  - vypsání nového předmětu – vypsání nového předmětu a přidělení správce
  - update předmětu – můžeme změnit jakýkoliv sloupec, ale musíme brát zřetel na studijní materiály – studijní materiály jsou uloženy jako zkratka\_předmětu.php (o studijních materiálech je odstavec níže)
  - odstranit předmět – při této akci se odstraní data i z tabulek spravuje a studuje
- Titulní stránka – v této sekci se upravují jednotlivé položky na titulní stránce kromě E-Přihlášky ke studiu.

## 5.2 Některé zajímavé funkce programu

Zde jsem zařadil některé funkce informačního systému U3V, které se mi zdály svým způsobem zajímavé.

### 5.2.1 Ostré testy

Ostré testy slouží pro otestování znalostí studentů a udělení odpovídajícího počtu bodů. Pro vytvoření testů jsem použil knihy [6], [7], [8] a [15].

Popis funkce:

1. **Zpřístupnění testu** – ještě před začátkem ostrého testu je nutné, aby správce předmětu zpřístupnil požadovaný test studentům. Potom, co všichni test zapnou, může jej klidně zase zablokovat, protože v průběhu ani po skončení testu se toto nastavení již nekontroluje.
2. **Kontrola před spuštěním** – před spuštěním testu se automaticky provede kontrola, jestli student test už dříve nepsal. Pokud má student v hodnocení za test 0 bodů, je vše v pořádku. Pokud má v hodnocení jakékoliv jiné bodové hodnocení, vyskočí hláška: „Na tento test nemáte již více pokusů!“. Těsně před spuštěním testu se totiž studentovi do hodnocení vybraného testu, který chce psát, zapíše -1 bod a následně se test spustí. Toto opatření je kvůli tomu, kdyby student test vypnul před jeho vyhodnocením.

3. **Spuštění testu** – test se spouští v novém okně, proto je potřeba povolit vyskakovací okna pro tuto stránku. Rozběhne se timeout. Odpočítávání do konce testu je řešeno pomocí JavaScriptové funkce a zobrazuje se do textového okénka se zablokováným psaním. Vygenerují se testové otázky. Zjistíme maximální číslo otázky, vypočteme náhodné číslo pomocí funkce `mt_rand ( 1 , $max )`. Postupně se generují náhodná čísla a zároveň se kontroluje, aby nebyla žádná dvě čísla stejná → pokud je nové číslo stejné jako již dříve vygenerované, zavoláme znovu funkci `mt_rand ( 1 , $max )`. Pokud již víme, že číslo je jedinečné, uložíme jej do pole čísel otázek. Až je vygenerováno právě tolik čísel, kolik má být v testu otázek, přejde se k další části. Postupně vybíráme vygenerovaná čísla otázek a vyhledáváme k nim v databázi text otázky, možnosti A-E a obtížnost. Vypočte se hodnota = body za test / součet obtížností za všechny vygenerované otázky. Podle hodnoty se vypočítají postupně body ke všem otázkám. Body za otázku = hodnota \* obtížnost za otázku. Všechny vygenerované otázky jsou uloženy v jednom formuláři. Při zakliknutí odpovědi se z pole otázek zjistí aktuální pozice a stránka se jako-by posune na další otázku. V podstatě se při kliknutí na odpověď odešle formulář a odpověď se uloží.
4. **Vyplnění testu** – vyplňování testových otázek je velmi intuitivní, neboť se po zakliknutí odpovědi automaticky posouváme dolů k další otázce. Postupně student vyplní všechny odpovědi u všech otázek.
5. **Ukončení testu** – k vyhodnocení testu dojde buď na základě vypršení časového limitu, nebo po kliknutí na tlačítko „Vyhodnotit test“ dole na stránce s otázkami.
6. **Vyhodnocení testu** – po ukončení testu vyskočí potvrzovací okénko s oznámením konce testu. Po stisknutí „OK“ přejdeme ke stránce s vyhodnocením. Při této akci je nutné odeslat pole s čísly otázek, pole s odpověďmi a hodnotu. Prochází se postupně obě pole, kontroluje se správnost odpovědí a vypisují se výsledky. Vypisuje se text otázky, zvolená odpověď, hodnocení k otázce – buď: „Zvolená odpověď byla správná, získáváte x bodů!“, nebo „Zvolená odpověď byla špatná!“ a k tomu se vypíše správná odpověď. Potom, co se vypíše hodnocení všech otázek, uloží se výsledek testu. Výsledek je součet získaných bodů zaokrouhlených nahoru. Pokud je výsledek 0, zapíše se -1 a pokud je výsledek vyšší než maximum za test, uloží se maximum za test. Nyní se musí přepočítat celkový počet bodů, vyhodnocení zápočtu a výsledná známka. Po prohlédnutí výsledku, můžeme stránku ukončit.

## 5.2.2 Cvičné testy

Tento typ testů je přístupný pro všechny studenty, kteří mají zapsaný konkrétní předmět. U cvičných testů stačí vybrat test a můžeme začít.

Pro spuštění není třeba žádné kontroly. Pokud student bude chtít, může si cvičné testy procházet i po napsání ostrého testu. Aby se studenti neučili pouze pomocí cvičného testu, jsou zde pouze nejjednodušší otázky s obtížností 1. Otázky se rovněž vybírají náhodně.

Princip funkce:

1. Náhodně se vygeneruje číslo otázky s obtížností 1.
2. Zvolíme některou z možných odpovědí.
3. Zobrazí se výsledek odpovědi a tlačítko pro přechod na další otázku. Pokud je odpověď správná, zobrazí se: „Zvolená odpověď je správná“, nebo naopak: „Zvolená odpověď je špatná!“.
4. Odpověď se může libovolně měnit, aby student mohl nalézt správnou odpověď.
5. Na další otázku se student dostane kliknutím na tlačítko pod otázkou.

Otázky se více nebo méně opakují, protože se zde nekontrolují čísla generovaných otázek. Pro tento typ testu není tato funkce potřebná.

### 5.2.3 Přihlašování uživatelů

Po zadání přihlašovacích údajů do přihlašovacího formuláře na hlavní stránce a jejich odeslání se musí zkontrolovat, kdo se chce přihlásit do systému. Kontroluje se:

1. **Přihlašovací jméno** – zjistíme, zda se nachází toto přihlašovací jméno v databázi. Pokud ne, tak uživatele nemůžeme pustit do systému. Na hlavní stránku se vypíše:

**Neznámé přihlašovací jméno!!!**

**Možné příčiny:**

**1) nejste v databázi doposud zaregistrováni**

**2) zadali jste svoje přihlašovací jméno špatně**

2. **Heslo** – přihlašovací jméno bylo nalezeno v databázi a rovnou jsme si k němu stáhli odpovídající heslo. Pokud heslo souhlasí, pokračuje se v přihlašovacím skriptu, jinak se vypíše: **Heslo je ŠPATNĚ nebo ŠPATNĚ přihlašovací jméno!!!**
3. **Aktivace** – kontrolujeme, zda má student aktivovaný účet. Účet není aktivovaný, pokud se nový student registruje přes E-Přihlášku na hlavní stránce. Aktivaci těmto studentům provede správce systému po úspěšném zvládnutí přijímacího řízení. Pokud je účet neaktivní, vypíše se: **Váš uživatelský účet nebyl zatím aktivován správcem!!!**

Nyní jsme zjistili, že uživatel je oprávněný ke vstupu do informačního systému U3V, resp. přihlašovací údaje odpovídají požadavkům a je třeba určit oprávnění konkrétního uživatele. Určujeme tyto typy oprávnění:

1. **Student** – studentům se načte stránka s přehledem studia – výpis předmětů.

2. **Správce předmětu** – první stránka, která se zobrazí, je výběr předmětu. Který bude správce spravovat.
3. **Správce systému** – základní stránka pro tyto uživatele je výběr oprávnění – pro přihlášení jako správce předmětu nebo celého systému.

## 5.2.4 Studijní materiály

Studijní materiály vytváří správce předmětu. Pro studenty jsou tyto materiály automaticky k dispozici po vybrání předmětu, v sekci studijní materiály. Pro vytvoření studijních materiálů jsem použil knihy [6], [7], [8] a [15].

Studijní materiály se upravují po zvolení předmětu v sekci studijní materiály. Po kliknutí na toto tlačítko, se správci zobrazí editační textové pole s obsahem souboru pro vybraný předmět. Každý předmět má pro tento účel automaticky vytvořen svůj soubor.

Na disk se tyto soubory ukládají do složky studijni\_materialy pod názvem zkratka\_předmětu.php. Vytvoření souboru proběhne automaticky při prvním uložení materiálů. Díky tomu, že zkratka předmětu je unikátní, nemělo by docházet ke kolizím. Může však dojít „ke ztrátě“ materiálů při změně zkratky předmětu. Předpokládám však, že správce systému bude o této věci vědět a případně si uloží kopii materiálů a potom ji uloží do nového souboru. Při této změně se soubory automaticky nesmažou. Takže i po změně zkratky předmětu se na disku data uchovávají i pro případ, že se správce jen nechtěně při měnění atributů předmětu ukliknul. Při následné změně zkratky zpět se se souborem může opět pracovat. Pokud se předmět ruší, nedá se soubor smazat z uživatelského rozhraní, ale musel by se odstranit přímo z disku, kde je uložen. Při rušení předmětu může správce předmětu vymazat obsah tohoto souboru, takže nebude na disku zabírat téměř žádné místo.

Po provedení editací se soubor ukládá kliknutím na tlačítko „Upravit materiály“. Aby měl správce představu, co přesně vytvořil, může použít funkci „Zobrazit náhled stránky“, která zobrazí výsledek pod textové pole.

Soubory s materiály se mohou vytvářet v HTML, což umožňuje provádět jednoduché formátování textu. Základy vytváření webových stránek by správci předmětu měli mít.

Pro vložené předměty jsem tyto materiály také vytvořil.

## 6 Testování systému

Testování systému všeobecně slouží k nalezení chyb a podle reakcí nezaujatých lidí se mohou měnit některé naprogramované funkce, případně vzhled aplikace.

Při testování jsem se zaměřil na použitelnost, funkčnost, možné díry při přihlašování do systému a na správnost všech naprogramovaných funkcí.

### 6.1 Testování použitelnosti

Pro zjištění správné funkčnosti a použitelnosti celého systému jsem testoval systém na několika subjektech. Úkolem testovaných osob bylo vyplnění E-Přihlášky ke studiu a následně se pokusit o vyplnění ostrých testů bez přečtení a po přečtení výukových materiálů. Tímto výzkumem jsem zjistil následující informace:

Mladší testované osoby zvládali přihlášku bez větších problémů. Na ostré testy jsem zvolil 10 minut, aby mohli i ti pomalejší stihnout projít všechny otázky. Počet otázek byl 20. Celkový maximální počet bodů za test také 20. Výsledek tohoto výzkumu je vyobrazen v [tabulce 1](#).

Jméno	Věk	Předmět	Čas (minuty)	Správné odpovědi	získané body
Jaroslav V.	47	ECOM	8	15	14
		H	6	15	15
		WXP	8,5	15	15
Miroslava H.	43	ECOM	7	13	13
		H	7,5	10	11
		WXP	6	15	14
Věra K.	38	ECOM	4	15	15
		H	2	15	15
		WXP	2,5	17	16
Klára S.	25	ECOM	5	12	11
		H	5	16	16
		WXP	7	16	17
Lukáš P.	18	ECOM	6	16	15
		H	5	19	19
		WXP	7	16	16
Průměr			5,77	15,00	14,80

Tabulka 1: Mladší generace bez přípravy



Druhou testovací skupinou byli starší lidé. S registrací přes elektronickou přihlášku se vyskytlo několik problémů, protože někteří před tímto výzkumem nikdy nepoužívali počítač. V [tabulce 2](#) je zobrazen výsledek ostrých testů pro tuto skupinu. Opět bylo v testu 20 otázek s maximem 20 bodů. Čas jsem prodloužil na 20 minut.

Jméno	Věk	Předmět	Čas (minuty)	Správné odpovědi	získané body
Jaroslav M.	67	ECOM	17	5	4
		H	18	11	11
		WXP	20	6	7
Věra M.	65	ECOM	10	14	14
		H	8	16	15
		WXP	13	9	9
Eva V.	72	ECOM	16	6	6
		H	16	9	9
		WXP	15	6	5
Petra S.	58	ECOM	9	5	4
		H	19	5	6
		WXP	8	9	10
Šimon S.	61	ECOM	5	14	15
		H	8	13	13
		WXP	8	16	15
Průměr			12,67	9,60	9,53

**Tabulka 2: Starší lidé bez přípravy**

Posledním testem byl pokus, kdy jsem této skupině nechal na tři dny vytištěné studijní materiály. Jak jsem již neznal, někteří nemají doma počítač, tak takhle měli všichni stejné podmínky. Před testem si ještě zájemci mohli 10 minut procházet cvičný test. Této možnosti nakonec využili všichni. Podmínky byly stejné jako pro první pokus. Výsledky jsou v [tabulce 3](#).

Výsledek výzkumu dopadl podle mého očekávání. I když mladší generace neměla k dispozici studijní materiály, testy jim nedělaly větší problémy. Naopak někteří starší lidé mohli bez přípravy jen tipovat. Po přečtení studijních materiálů, byly již výsledky o něco lepší.

Jméno	Věk	Předmět	Čas (minuty)	Správné odpovědi	získané body
Jaroslav M.	67	ECOM	12	18	17
		H	10	12	12
		WXP	17	15	14
Věra M.	65	ECOM	10	14	13
		H	9	16	15
		WXP	13	12	11
Eva V.	72	ECOM	16	8	7
		H	14	7	7
		WXP	15	11	12
Petra S.	58	ECOM	8	8	7
		H	13	9	10
		WXP	9	12	12
Šimon S.	61	ECOM	3	17	16
		H	5	14	13
		WXP	5	15	14
Průměr			10,60	12,53	12,00

Tabulka 3: Starší lidé s přípravou

## 6.2 Testování přihlašování

Kromě těchto testů jsem se zkoušel přihlašovat bez přihlašovacího jména a hesla, nebo se špatnými údaji. Systém na tato nesprávná přihlášení reagoval správně.

## 6.3 Testování správné funkčnosti

Při testování funkčnosti jsem narazil na několik nesprávně fungujících funkcí, které jsem po jejich odhalení opravil. Nyní by měl být informační systém připraven k použití v praxi.

## 7 Závěr

Závěrem své bakalářské práce bych chtěl říci, že programování tohoto projektu mě obohatilo o nové informace a zkušenosti, které doufám, že budu moci uplatnit v budoucnu při vytváření nových informačních systémů.

Kromě nových informací z oblasti programování, jsem se dozvěděl i mnoho zajímavých věcí při vytváření testů a sepisování studijních materiálů.

Doufám, že moje práce přinese budoucím uživatelům této webové aplikace vhodné uživatelské rozhraní pro správu systému a studentům dobré znalosti, které uplatní v praktickém životě při práci s počítačem, používání internetu či e-mailu.

Možným rozšířením by mohlo být zobrazení grafu s úspěšností všech studentů. Každý předmět by měl svůj graf úspěšnosti celkově i v jednotlivých testech. Tyto grafy by se mohly nacházet na osobních stránkách každého studenta.

# Použité zkratky

CSS – Cascading Style Sheets

DTD – Definice Typu Dokumentu

ECMA – European Computer Manufacturers Association

ER – Entity Relationship

HTML – Hypertext Markup Language

ISO – International Organization for Standardization

LAMP – Linux, Apache, MySQL a PHP nebo Perl

NF – normální forma

OS – Operační Systém

PHP – dříve Personal Home Page, nyní Hypertext Preprocessor

SGML – Standard Generalized Markup Language

SQL – Standard Query Language

U3V – Univerzita třetího věku

W3C – The World Wide Web Consortium

WWW – World Wide Web

# Literatura

- [1] LINHART, P.: *Apache & PHP*. Praha: Neokortex spol. s.r.o., 1998, ISBN 80-902230-2-8.
- [2] LEE, J., WARE, B.: *Open Source – vývoj webových aplikací*. Praha: Mobil Media a.s., 2003, ISBN 80-86593-43-6.
- [3] PÍSEK, S.: *HTML a XHTML*. Praha: Grada Publishing a.s., 2003, ISBN 80-247-0571-0.
- [4] STANÍČEK, P.: *CSS Kaskádové styly: Kompletní průvodce*. Brno: Computer Press a.s., druhé vydání, 2003, ISBN 80-7226-872-4.
- [5] DELLWIG, E., DELLWIG, I.: *JavaScript příručka programátora*. Praha: Grada Publishing a.s., 2003, ISBN 80-247-0298-3.
- [6] RYBKA, M., VLČEK, V.: *Hardware: Jak porozumět svému počítači*. Praha: Grada Publishing a.s., 2002, ISBN 80-247-0151-0.
- [7] LAPÁČEK, J.: *Počítač v domácnosti*. Brno: Computer Press a.s., 2006, ISBN 80-251-0945-3.
- [8] LAPÁČEK, J.: *Počítač pro seniory*. Brno: Computer Press a.s., 2003, ISBN 80-7226-810-4.
- [9] *JavaScript*. [online]. Wikipedie, poslední úpravy 30. 10. 2008 [cit. 2008-11-04]. Dostupné na WWW: <<http://cs.wikipedia.org/wiki/JavaScript>>.
- [10] TACHOVSKÁNET. *Skriptovací jazyk PHP*. [online]. Poslední úpravy 12. 7. 2008 [cit. 2008-11-12]. Dostupné na WWW: <<http://www.tachovska.cz/server/php.php>>.
- [11] *PHP*. [online]. Wikipedie, poslední úpravy 20. 10. 2008 [cit. 2008-11-12]. Dostupné na WWW: <<http://cs.wikipedia.org/wiki/Php>>.
- [12] *MySQL*. [online]. Wikipedie, poslední úpravy 28. 10. 2008 [cit. 2008-11-12]. Dostupné na WWW: <<http://cs.wikipedia.org/wiki/MySQL>>.
- [13] *CSS*. [online]. Wikipedie, poslední úpravy 23. 12. 2008 [cit. 2009-01-03]. Dostupné na WWW: <[http://cs.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](http://cs.wikipedia.org/wiki/Cascading_Style_Sheets)>.
- [14] VEBLOUD. *Teorie relačních databází*. [online]. Poslední editace 2. 8. 2007 [cit. 2009-01-08]. Dostupné na WWW: <<http://www.manualy.net/article.php?articleID=13>>.
- [15] MICROSOFT CORPORATION: *Microsoft Windows XP Professional Resource Kit*. Brno: Computer Press a.s., 2004, ISBN 80-7226-608-X.

# Seznam příloh

- CD se zdrojovými kódy

## Obsah CD

- Složka src – obsahuje všechny vytvořené zdrojové soubory aplikace
- Readme.txt – základní informace o aplikaci, přihlašovací jména a hesla, umístění informačního systému na internetu
- Install.sql – soubor, obsahující skript pro vytvoření tabulek a vložení ukázkových dat
- Bp.pdf – elektronická verze textu bakalářské práce